

## COMMUNITY

Arcia Osorio Jairo Andrés  
Sierra Oliveros Sara Margarita  
Solano Romero Jorge Junior  
Villa Bastidas Daniela Isabel



UNIVERSIDAD LIBRE  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA DE SISTEMAS  
BARRANQUILLA / ATLÁNTICO  
2024-2

## COMMUNITY

Arcia Osorio Jairo Andrés  
Sierra Oliveros Sara Margarita  
Solano Romero Jorge Junior  
Villa Bastidas Daniela Isabel

### PRESENTACIÓN DE PROYECTO, COMMUNITY - CENTRO DE BÚSQUEDA Y COLABORACIÓN EN AYUDA GRATUITA

Docentes;  
Berrio Rodríguez Emiro  
De la Hoz Ricardo  
Sanmartín Mendoza Paul



UNIVERSIDAD LIBRE  
FACULTAD DE INGENIERÍA  
PROGRAMA INGENIERIA DE SISTEMAS  
BARRANQUILLA / ATLANTICO  
2024-2

## Contenido

<b>1. INTRODUCCIÓN .....</b>	<b>4</b>
<b>2. OBJETIVOS.....</b>	<b>4</b>
<b>2.1 Objetivo General .....</b>	<b>4</b>
<b>2.2 Objetivos Específicos.....</b>	<b>4</b>
<b>3. PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>5</b>
<b>3.1 Definición del Problema.....</b>	<b>5</b>
<b>3.2 Justificación .....</b>	<b>5</b>
<b>4. MARCO TEÓRICO .....</b>	<b>5</b>
<b>4.1 Estilos de Arquitectura.....</b>	<b>5</b>
<b>4.2 ODS 17: Alianzas para lograr los Objetivos .....</b>	<b>6</b>
<b>4.3 Bases de Datos Relacionales .....</b>	<b>7</b>
<b>5. MATERIALES Y METODOLOGÍA .....</b>	<b>7</b>
<b>5.1 Materiales.....</b>	<b>7</b>
<b>5.2 Metodología.....</b>	<b>7</b>
<b>6. DESARROLLO DEL PROYECTO .....</b>	<b>8</b>
<b>6.1 Análisis del Desarrollo del Proyecto.....</b>	<b>8</b>
<b>6.2 Cronograma .....</b>	<b>8</b>
<b>7. CONCLUSIONES.....</b>	<b>9</b>
<b>8. RECOMENDACIONES.....</b>	<b>9</b>
<b>9. BIBLIOGRAFÍA.....</b>	<b>10</b>

# 1. INTRODUCCIÓN

El proyecto **CommUnity** tiene como objetivo centralizar el acceso a recursos de ayuda gratuita, facilitando la colaboración y el apoyo entre personas y organizaciones. Este proyecto contribuye al **Objetivo de Desarrollo Sostenible (ODS) 17: Alianzas para lograr los Objetivos**, al proporcionar una plataforma digital que permite a individuos y comunidades acceder a servicios de ayuda, fortaleciendo las alianzas entre plataformas existentes.

## 2. OBJETIVOS

### 2.1 Objetivo General

Desarrollar una plataforma distribuida y centralizada que facilite el acceso a recursos de ayuda gratuita, promoviendo alianzas entre plataformas de asistencia y contribuyendo al ODS 17 mediante el fortalecimiento de la colaboración entre diferentes entidades y organizaciones.

### 2.2 Objetivos Específicos

1. **Crear una Interfaz de Usuario Intuitiva:** Facilitar la búsqueda de ayuda y el registro de voluntarios, permitiendo a los usuarios acceder fácilmente desde dispositivos móviles y de escritorio.
2. **Integrar Servicios de Ayuda Externos mediante APIs:** Conectar plataformas de ayuda reconocidas como Make-A-Wish y Eduteam, alineándose con el ODS 17 al fomentar alianzas estratégicas.
3. **Implementar una Base de Datos Relacional:** Utilizar MySQL para gestionar eficientemente la información de usuarios y solicitudes, garantizando integridad y escalabilidad.

4. **Desarrollar un Sistema de Notificaciones y un Panel Administrativo:** Alertar a los usuarios sobre nuevas oportunidades de ayuda y facilitar la gestión de usuarios y recursos a través de un panel administrativo.
5. **Garantizar la Seguridad de la Información del Usuario:** Aplicar estándares de cifrado y autenticación para proteger la información personal de los usuarios.

### 3. PLANTEAMIENTO DEL PROBLEMA

#### 3.1 Definición del Problema

La falta de una plataforma unificada que centralice el acceso a recursos de ayuda gratuita dificulta la colaboración entre individuos y organizaciones. Este problema impide la creación de alianzas estratégicas que puedan amplificar el impacto de los recursos de ayuda, limitando el alcance de las plataformas de asistencia disponibles.

#### 3.2 Justificación

El proyecto **CommUnity** responde a esta necesidad al centralizar la oferta de ayuda en una plataforma accesible, promoviendo el ODS 17 mediante la integración de recursos de múltiples plataformas y facilitando el acceso a una red colaborativa.

### 4. MARCO TEÓRICO

#### 4.1 Estilos de Arquitectura

**CommUnity** se construye sobre una **arquitectura de microservicios**, un enfoque que permite diseñar aplicaciones como una colección de servicios independientes. Cada microservicio ofrece una funcionalidad específica, como la búsqueda de ayuda, el manejo de

usuarios o el sistema de notificaciones. Las características principales de esta arquitectura incluyen:

- **Despliegue en contenedores con Docker:** Los microservicios se empaquetan en contenedores Docker, lo que facilita su despliegue en cualquier entorno y su escalabilidad de manera autónoma. Esto permite que cada servicio se ejecute independientemente, brindando flexibilidad en el mantenimiento y actualizaciones.
- **Orquestación con Kubernetes:** Kubernetes se utiliza para gestionar y orquestar los contenedores, permitiendo un balance de carga óptimo y garantizando alta disponibilidad. Este sistema permite la escalabilidad automática, lo que asegura que la plataforma puede manejar incrementos en el tráfico sin comprometer el rendimiento.
- **Patrón de Diseño MVC (Modelo-Vista-Controlador):** El diseño MVC organiza el sistema en tres componentes principales:
  - **Modelo:** Gestiona la lógica de negocio y el acceso a la base de datos.
  - **Vista:** Define la interfaz de usuario en Blazor, optimizando la experiencia de usuario en dispositivos móviles y de escritorio.
  - **Controlador:** Interpreta las entradas del usuario, enviándolas al modelo para el procesamiento y luego regresando la vista correspondiente.

## 4.2 ODS 17: Alianzas para lograr los Objetivos

El ODS 17 busca fomentar alianzas estratégicas para abordar problemas de desarrollo sostenible. **CommUnity** contribuye a este objetivo al ofrecer una plataforma donde diversas entidades y personas pueden unirse en torno a una causa común de ayuda gratuita. Mediante la integración de plataformas de ayuda y la centralización de recursos, **CommUnity** facilita el acceso a redes de apoyo y refuerza las asociaciones digitales, maximizando el impacto positivo en las comunidades.

### 4.3 Bases de Datos Relacionales

El sistema se apoya en una base de datos relacional (MySQL) para organizar la información de los usuarios y los recursos de ayuda. Las bases de datos relacionales aseguran:

- **Gestión Eficiente de Datos:** La estructura relacional permite gestionar la información de usuarios, voluntarios y solicitudes de ayuda, manteniendo la integridad y facilitando la recuperación de datos de manera eficiente.
- **Integridad de Datos:** Gracias a los controles de integridad referencial, la información almacenada es consistente y precisa, mejorando la confiabilidad de los datos.
- **Escalabilidad:** El diseño relacional facilita el crecimiento del sistema, ya que puede expandirse para incluir nuevas funcionalidades y usuarios sin comprometer la estabilidad.

## 5. MATERIALES Y METODOLOGÍA

### 5.1 Materiales

- **Lenguajes y Herramientas:** ASP.NET Core y Blazor para desarrollo web, Docker para contenerización y Kubernetes para orquestación.
- **Base de Datos Relacional:** MySQL para gestionar datos estructurados y consultas eficientes.

### 5.2 Metodología

El desarrollo se gestionará bajo la metodología **Scrum**, que permite un proceso iterativo y adaptable a cambios mediante sprints de dos semanas, con revisiones y planificación constante para asegurar la entrega continua de valor.

## 6. DESARROLLO DEL PROYECTO

### 6.1 Análisis del Desarrollo del Proyecto

1. **Diseño de Interfaz de Usuario:** Una experiencia fluida y accesible en dispositivos móviles y de escritorio.
2. **Integración de APIs:** Conectar plataformas externas de ayuda mediante APIs para consolidar recursos.
3. **Gestión de Base de Datos:** La base de datos MySQL almacena datos de usuarios, solicitudes y ofrece integridad relacional.
4. **Sistema de Notificaciones:** Mantiene a los usuarios informados sobre nuevas oportunidades de ayuda.
5. **Panel Administrativo:** Herramientas de gestión para administradores y generación de reportes.

### 6.2 Cronograma

Fase ciclo de vida del software	Duración
Análisis de Requisitos	1 – 2 Semanas
Diseño de Arquitectura y Base de Datos	1 – 2 semanas
Desarrollo e Integración de Microservicios	Por definir
Pruebas y Validación	Por definir
Despliegue y Mantenimiento	Por definir



## 7. CONCLUSIONES

**CommUnity** se proyecta como una solución eficaz para centralizar el acceso a la ayuda gratuita y fortalecer las alianzas, contribuyendo así al ODS 17. La arquitectura de microservicios asegura un rendimiento escalable, mientras que la base de datos relacional garantiza la integridad de la información, facilitando un entorno digital de colaboración confiable y sostenible.

## 8. RECOMENDACIONES

- **Actualización Continua de APIs:** Mantener las conexiones actualizadas para que la información sea precisa y útil.
- **Optimización de Seguridad y Rendimiento:** A medida que crece el sistema, mejorar los protocolos de seguridad y rendimiento de la base de datos.
- **Ampliación de Alianzas:** Explorar y desarrollar alianzas con nuevas plataformas de ayuda para expandir el alcance del sistema.

## 9. BIBLIOGRAFÍA

Pressman, R. S. Ingeniería de Software. McGraw-Hill, 1995.

Sommerville, I. Ingeniería de Software. Pearson Educación, 2011.

Silberschatz, A., Korth, H. F., & Sudarshan, S. Fundamentos de bases de datos. 6ª ed. México: McGraw-Hill, 2014.

Naciones Unidas. Objetivo de Desarrollo Sostenible 17: Alianzas para lograr los Objetivos. <https://www.un.org/sustainabledevelopment/es/globalpartnerships/>.

IEEE. IEEE Standard 830-1998. IEEE Recommended Practice for Software Requirements Specifications. The Institute of Electrical and Electronics Engineers, 1998.